

**ANÁLISIS Y DISEÑO DE APLICATIVO MÓVIL PARA LA GESTIÓN DE RUTAS
Y HORARIOS DE LOS BUSES EN LA CIUDAD PEREIRA**



Universidad
Tecnológica
de Pereira

**JUAN MIGUEL VARGAS CORTÉS
JUAN PABLO BOTERO BERMÚDEZ**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
FACULTAD DE TECNOLOGÍAS
PEREIRA
2021**

**ANÁLISIS Y DISEÑO DE APLICATIVO MÓVIL PARA LA GESTIÓN DE RUTAS
Y HORARIOS DE LOS BUSES EN LA CIUDAD PEREIRA**

**JUAN MIGUEL VARGAS CORTÉS
JUAN PABLO BOTERO BERMÚDEZ**

Proyecto de grado
Para optar por el título de Ingeniería en Sistemas y Computación
e Ingeniería en Mecatrónica

Directores:
ING. CÉSAR MANUEL CASTILLO RODRÍGUEZ
PhD. EDUARDO GIRALDO SUÁREZ

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
FACULTAD DE TECNOLOGÍAS
PEREIRA
2021**

NOTA DE ACEPTACIÓN

FIRMA DIRECTOR

FIRMA JURADO

FIRMA JURADO

AGRADECIMIENTOS

En primer lugar, dar las gracias a la Universidad Tecnológica de Pereira por permitir nuestra formación académica y profesional, agradecer a todos los maestros que hicieron parte de nuestra formación y aportaron en la creación de las personas que somos hoy en día, y en especial al ingeniero César Manuel Castillo Rodríguez y al Doctor Eduardo Giraldo Suárez por su acompañamiento y guía en la realización de este trabajo de grado. Agradecer a Dios por permitirnos vivir esta aventura de gran aprendizaje; por último y no menos importante, a nuestras familias que han sido un gran aliciente para culminar esta meta, por su apoyo incondicional en estos años y su confianza durante todo el proceso.

CONTENIDO

RESUMEN	7
INTRODUCCIÓN	9
1. PROBLEMA DE INVESTIGACIÓN	10
1.1. PLANTEAMIENTO DEL PROBLEMA	10
1.2. FORMULACIÓN DEL PROBLEMA	10
1.3. SISTEMATIZACIÓN DEL PROBLEMA	10
1.4. DELIMITACIÓN	11
2. JUSTIFICACIÓN	12
3. OBJETIVOS	13
3.1. OBJETIVO GENERAL	13
3.2. OBJETIVOS ESPECÍFICOS	13
4. MARCO DE REFERENCIA	14
4.1. MARCO TEÓRICO	14
4.2. MARCO CONCEPTUAL	18
4.3. MARCO ESPACIAL	19
4.4. MARCO TEMPORAL	20
5. METODOLOGÍA	21
5.1. TIPO DE INVESTIGACIÓN	21
6. DESARROLLO DEL PROYECTO	22
6.1. APLICACIÓN DESARROLLADA	24
6.2. DISPOSITIVO DE VISUALIZACIÓN	25
6.2.1. Programa ARDUINO	25
6.2.2. Programa MATLAB	26
6.3. ENCUESTA	27
6.4. PRUEBAS	30
7. CONCLUSIONES	31
8. RECOMENDACIONES	32
BIBLIOGRAFÍA	33

Lista de ilustraciones

Ilustración 1. Cronograma de actividades

Ilustración 2. UML Prototipo Bus a Tiempo

Ilustración 3. Caso de Uso Bus a Tiempo

Ilustración 4. Fórmula para hallar muestra en población finita

Ilustración 5. Resultado de la fórmula de muestra en población finita

Ilustración 6. Respuestas de la pregunta número 1

Ilustración 7. Respuestas de la pregunta número 2

Ilustración 8. Respuestas de la pregunta número 3

Ilustración 9. Respuestas de la pregunta número 4

Anexos

ANEXO A. Aplicación Desarrollada

ANEXO B. Dispositivo de visualización

ANEXO C. Encuesta

ANEXO D. Videos

ANEXO E. Tabla de pruebas

ANEXO F. Respuesta a la solicitud de información de la empresa MEGABÚS S.A.

RESUMEN

El trabajo de grado titulado “ANÁLISIS Y DISEÑO DE APLICATIVO MÓVIL PARA LA GESTIÓN DE RUTAS Y HORARIOS DE LOS BUSES EN PEREIRA” es un estudio enfocado en desarrollar un sistema de información para la consulta de rutas y horarios de los buses públicos de la ciudad de Pereira, donde los usuarios de este servicio, pueden consultar su transporte de interés a través de un aplicativo móvil en su celular e igualmente en pantallas o dispositivos de visualización que se ponen a disposición en los paraderos o puntos de interés.

Se evidencia que una mejor gestión del tiempo de espera del transporte público puede ayudar significativamente a la satisfacción y experiencia de los usuarios y brindar mejor calidad en la prestación de este servicio.

Así, por medio de un aplicativo móvil disponible en el celular de los usuarios y con un vistazo rápido, podrán consultar la ruta y el tiempo que tarda en llegar el transporte de su interés.

ABSTRACT

The degree work entitled "ANALYSIS AND DESIGN OF MOBILE APPLICATION FOR THE MANAGEMENT OF ROUTES AND SCHEDULES OF BUSES IN PEREIRA" is a study focused on developing an information system for consulting routes and schedules of public buses in the city of Pereira, where users of this service can consult their transport of interest through a mobile application on their cell phone and also on screens or display devices that are available at stops or points of interest.

It is evident that better management of public transport waiting time can significantly improve user satisfaction and experience and provide better quality in the provision of this service

So, through a mobile application available on users' cell phones and with a quick glance, they will be able to check the route and the time it takes for the transport of their interest to arrive

INTRODUCCIÓN

El sostenimiento y éxito de un servicio, producto o empresa a lo largo del tiempo se basa en gran medida en la calidad y aporte de valor que pueda brindar a sus usuarios o clientes. En empresas prestadoras de servicios, la experiencia de usuario tiene gran relevancia en sus objetivos y metas.

Las empresas prestadoras de servicio de transporte público tienen en su misión operar bajo condiciones de seguridad, confiabilidad, eficiencia y economía; Garantizando su sostenibilidad, responsabilidad social y ambiental, contribuyendo al mejoramiento de la calidad de vida de su población.

De ahí que el mejoramiento de la calidad de vida sea de gran importancia en la actualidad. Esto se ha conseguido gracias a grandes avances tecnológicos, científicos, sociales, económicos, en infraestructura, en procesos y muchos más.

En los últimos años los avances más significativos o relevantes se han dado en el ámbito tecnológico, pues permite facilitar en gran medida las tareas diarias, sean en la vida personal o laboral. Aplicativos móviles que se pueden llevar en teléfonos celulares y están disponibles con mucha facilidad en cualquier momento del día; tener la capacidad de comunicarse con los demás en un instante, realizar consultas por internet, encontrar un sitio de interés y muchas tareas más que están a nuestro alcance gracias a estos avances.

Por consiguiente, se busca aprovechar esta tecnología para brindar un mejor servicio y un valor agregado a los usuarios de transporte público. Así, el actual trabajo de grado que se presenta como requisito para optar por los títulos de Ingeniera en Sistemas y Computación e Ingeniería en Mecatrónica, se basa en torno al desarrollo de un sistema de información para la consulta de rutas y horarios de los buses públicos de la ciudad de Pereira.

Este proyecto se elabora con datos derivados de una investigación propia por parte de los autores, donde se consultan distintas fuentes de información que tienen relación con los temas antes mencionados y otros que se ampliarán más adelante en este documento. Se precisa un buen manejo de la información, buen conocimiento en el área para el desarrollo del proyecto y un pensamiento lógico en la construcción del aplicativo.

1. PROBLEMA DE INVESTIGACIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

El tiempo es un elemento valioso que se debe gestionar de manera efectiva para evitar desaprovecharlo en actividades poco relevantes o poco productivas; la pérdida de tiempo es inadmisibles en tiempos actuales donde se lleva un estilo de vida muy acelerado y donde cada minuto cuenta.

Esto les sucede a los usuarios de transporte público en una parada de autobús mientras tienen la incertidumbre del momento exacto de la llegada de su transporte.

En el presente, la calidad es el fundamento en la evolución de la prestación de un servicio, pues entre más se avanza, más exigencias se generan con el aprovechamiento de los recursos y el tiempo.

1.2. FORMULACIÓN DEL PROBLEMA

Esta situación lleva a plantear el siguiente interrogante, ¿Una aplicación móvil para observar horarios de llegada de un autobús público, mejorará la gestión de tiempo de un usuario final, respecto a las actividades desempeñadas en su diario vivir?

1.3. SISTEMATIZACIÓN DEL PROBLEMA

- ¿Se cuenta en la actualidad con un sistema de información, para la consulta de rutas y horarios de buses en Pereira, funcional y confiable?
- ¿Qué información con la que se cuenta es útil para el desarrollo del sistema de información?
- ¿Es viable el desarrollo e implementación de un sistema de información para la consulta de rutas y horarios de buses en Pereira?

1.4. DELIMITACIÓN

El proyecto tiene como alcance desarrollar un sistema de información para la consulta de rutas y horarios de los buses públicos de la ciudad de Pereira. En este caso se diseñará un aplicativo móvil funcional para la consulta de una ruta de transporte público en específico. Esto servirá como base para replicar con las demás rutas de interés pero que no están dentro del alcance de este proyecto. Todo esto servirá para el beneficio del usuario y el enriquecimiento del servicio por parte de las empresas prestadoras del servicio de transporte público.

2. JUSTIFICACIÓN

Un sistema de información es un conjunto de datos que interactúan entre sí con un fin común. Los sistemas de información ayudan a administrar, recolectar, recuperar, procesar, almacenar y distribuir información relevante para los procesos fundamentales y las particularidades de cada organización.

La importancia de un sistema de información radica en la eficiencia en la correlación de una gran cantidad de datos, ingresados a través de procesos diseñados para cada área, con el objetivo de producir información válida para la posterior toma de decisiones.

A nivel mundial se tienen sistemas y aplicativos de este tipo para la consulta de rutas y horarios de diferentes medios de transporte público, donde se evidencia su confiabilidad y aumento, tanto en la calidad del servicio, como en la satisfacción de sus usuarios.

Con un sistema de información como base fundamental del proyecto, se busca cubrir una necesidad presente en el día a día de los usuarios de transporte público en la ciudad de Pereira, y así, reducir la incertidumbre y mejorar de manera sustancial la gestión del tiempo de espera del transporte de su interés.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Desarrollar un sistema de información para la consulta de rutas y horarios de los buses de la ciudad de Pereira.

3.2. OBJETIVOS ESPECÍFICOS

- Recopilar información, organizarla de forma detallada y delimitar el alcance y los requerimientos del sistema.
- Investigar las opciones más viables para desarrollar la aplicación por medio de lenguajes de programación, frameworks y sistemas y elementos para la geolocalización de los buses.
- Desarrollar un prototipo funcional de la aplicación móvil.
- Desarrollar un prototipo de dispositivo de visualización que permita ver, al usuario, el tiempo de espera de su transporte de interés.

4. MARCO DE REFERENCIA

4.1. MARCO TEÓRICO

Este tipo de aplicativo no es algo nuevo, pues ya existen en el mundo varias aplicaciones sobre la movilidad en las ciudades, por ejemplo: En Estados Unidos existe una aplicación donde se encuentra información en tiempo real de las vías, tiempos de espera y retrasos de los trenes, buses y tranvía, a esta aplicación se le conoce como MyTransit NYC subway, bus, rail (MTA). También se encuentra otra aplicación en tiempo real, la cual está operativa en cuatro países (España, Canadá, Estados Unidos y México) llamada Transit Bus & Train Times, es un mapa similar a Google Maps donde se hallan las rutas de los buses y trenes con el tiempo de espera en tiempo real.

En la ciudad de Pereira se toma como referencia la empresa de MEGABÚS S.A., pues es claro que es muy importante en el área metropolitana y muy influyente en el transporte público.

En el momento de realizar este documento la empresa no cuenta con ningún aplicativo para la consulta de rutas y horarios de sus buses y ningún recurso bien implementado o completamente funcional que indique el tiempo de espera para sus buses o alimentadores.

Sin embargo, los usuarios del sistema MEGABUS pueden consultar las rutas (alimentadoras y troncales) y recorridos, en la página de la entidad www.megabus.gov.co. (*Ver anexo F*)

“Un sistema de información es un conjunto de datos que interactúan entre sí con un fin común.”

Los sistemas de información surgen de la necesidad de organizar y administrar recursos y por lo tanto son tan antiguos como la civilización misma. Para conocer el origen de estos sistemas se remonta hasta la civilización egipcia (4000 AC.) para encontrar el uso de los censos de población. Un censo se encarga de recoger información, procesarla y utilizarla para la toma de decisiones, por lo tanto, se puede considerar como sistema de información.

Los sistemas informáticos han ido mejorando desde entonces, aunque es desde mediados del siglo pasado hasta hoy en día cuando más han cambiado.

Durante la década de los 50's, los sistemas de información eran todavía muy simples debido a la tecnología de la época y su principal función era servir de ayuda a los trabajadores realizando diferentes tipos de tareas como puede ser procesar

transacciones, mantener registros o llevar la contabilidad. Estos sistemas son conocidos como TPS o sistemas de procesamiento de transacciones. Un ejemplo de ello serían los cajeros automáticos.

En los 60's surgen los sistemas de información gerenciales o MSI que tienen la novedad de transformar los datos almacenados en información útil para ayudar a las empresas en la toma de decisiones. Algunas cosas que son capaces de hacer estos sistemas son: Generar informes detallados de transacciones como puede ser el informe de un pedido, generar informes de tendencias de costes o generar informes de análisis de ventas.

De 1970 a 1980 se produjo un avance importante con el surgimiento de las computadoras personales lo que facilitó la expansión de los sistemas informáticos a toda la organización.

Cabe destacar también los sistemas de soporte de decisiones o DSS. Estos sistemas interactivos toman datos de diversas fuentes tanto internas (inventario, ventas) como externas (tendencias del mercado, precio de mercado) y los transforman mediante complejos algoritmos en información útil para ayudar a la alta dirección de la empresa en la toma de decisiones.

En los 80's destacan los sistemas de información ejecutivo o ESI como una evolución de los DSS. Estos sistemas ayudan en la toma de decisiones a la ejecutiva con el fin de cumplir sus objetivos estratégicos.

En la época de los 90's el surgimiento de Internet cambió drásticamente las capacidades de los sistemas de información en los negocios ya que hizo posible intercambiar información en tiempo real con diferentes partes del mundo.

También surgieron los conocidos ERP (sistemas de planificación de recursos empresariales) que proporcionan información estratégica a departamentos de una organización (ventas, compras, recursos humanos etc).

Otros sistemas importantes que surgieron en esta década fueron los sistemas expertos o ES, capaces de imitar la toma de decisiones de los humanos haciendo uso de sistemas de gestión del conocimiento (KMS), sistemas inteligentes encargados de generar conocimiento a través de datos. Algunas cosas que un ES puede hacer es por ejemplo diagnosticar enfermedades y hacer predicciones financieras.

Por último, en los últimos años los sistemas de información no es que hayan cambiado en gran medida en cuanto al tipo de funcionalidades que ofrecían anteriormente, sino que han ido mejorando debido a los avances tecnológicos (mayor capacidad de almacenamiento, mejores infraestructuras de red, cloud computing, etc.).

Los sistemas de información están compuestos por cuatro componentes: tarea, personas, estructura (o roles) y tecnología. Los sistemas de información se pueden definir como una integración de componentes para la recopilación, almacenamiento y procesamiento de datos, y éstos se utilizan para proporcionar información, contribuir al conocimiento y productos digitales.

Los seis componentes que deben unirse para producir un sistema de información son:

Hardware: el término hardware se refiere a maquinaria. Esta categoría incluye la computadora en sí, y todo su equipo de soporte. Entre los equipos de soporte se encuentran dispositivos de entrada y salida, dispositivos de almacenamiento y dispositivos de comunicaciones.

Software: El término software se refiere a los programas de computadora y los manuales (si los hay) que los respaldan. Los programas de computadora son instrucciones legibles por máquina que dirigen los circuitos dentro de las partes de hardware del sistema para que funcionen de manera que produzcan información útil a partir de los datos.

Datos: Los datos son hechos que utilizan los programas para producir información útil. Al igual que los programas, los datos generalmente se almacenan en forma legible por máquina en disco o cinta hasta que la computadora los necesita.

Procedimientos: Los procedimientos son las políticas que gobiernan el funcionamiento de un sistema informático. "Los procedimientos son para las personas lo que el software es para el hardware" es una analogía común que se utiliza para ilustrar el papel de los procedimientos en un sistema.

Personas: Todo sistema necesita personas para ser útil. A menudo, el elemento del sistema que más se pasa por alto son las personas, probablemente el componente que más influye en el éxito o el fracaso de los sistemas de información.

Retroalimentación: Define que un sistema de información puede recibir retroalimentación.

En este prototipo de sistema de información se utiliza Firebase de Google como la capa de acceso a datos del software (Back-End), que es una plataforma en la nube para el desarrollo de aplicaciones web y móvil, la cual está disponible para plataformas tales como iOS, Android y Web, es sencilla de usar y con alta calidad.

Para la parte del programa a la que el usuario puede acceder directamente (Front-End), se emplea Angular en diferentes versiones, la cual trabaja con patrón de programación MVC (Modelo-Vista-Controlador) que es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos:

- *Modelo*: Contiene una representación de los datos que maneja el sistema, su lógica de negocio y sus mecanismos de persistencia.
- *Vista*: Es la interfaz de usuario, compone la información que se envía al cliente y los mecanismos de interacción con éste.
- *Controlador*: Actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

Angular, es un framework de Javascript de código abierto, mantenido por Google, que se utiliza para crear, mantener aplicaciones web en una sola página. La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Angular se basa en clases de tipo “componentes”, cuyas prioridades son las usadas para hacer el binding de los datos (enlace de datos, que enlaza los elementos de la interfaz con la parte en se arma la información a mostrar, lo que es un enlace entre el front y el back).

API (Application Programming Interface), básicamente es un conjunto de aplicaciones que permiten la construcción de una interfaz inteligente, configurando un medio por el cual dos sistemas se comunican.

Google Maps API, permite a los usuarios visualizar el mundo real, por medio de para estáticos o interactivos, y pueden ser personalizados e incorporados a sitios o aplicaciones.

API JavaScript de Maps, permite agregar mapas interactivos y personalizados con contenido propio e imágenes para exhibición en páginas web y dispositivos móviles.

Routes (Google Maps), ayuda a los usuarios a encontrar el mejor trayecto hasta su destino, con la sugerencia de rutas y actualizaciones de tránsito en tiempo real- Además de eso, también permite crear itinerarios hasta de 25 puntos de referencia, ideal para un planeamiento de rutas de entrega más eficiente.

API Directions, permite proporcionar rutas de transporte público, bicicleta, auto y a pie.

API Roads, posibilita crear itinerarios precisos, determinando el trayecto a ser recorrido, además de proporcionar datos al respecto de esas rutas, como límite de velocidad.

4.2. MARCO CONCEPTUAL

Sistema de información

Conjunto de datos que interactúan entre sí con un fin común. Los sistemas de información ayudan a administrar, recolectar, recuperar, procesar, almacenar y distribuir información relevante para los procesos fundamentales y las particularidades de cada organización.

Aplicación móvil

Es una aplicación informática diseñada para ser utilizada en dispositivos móviles que permite al usuario ejecutar una tarea en específico o gran variedad de ellas.

Gestión

Es llevar a cabo diligencias que hacen posible la realización de una operación comercial o de un anhelo cualquiera.

Transporte público

Es el transporte colectivo de pasajeros, los viajeros del transporte público tienen que adaptarse a los horarios y a las rutas que ofrezca el operador y dependen en mayor o menor medida de la intervención regulatoria del Gobierno.

Usuario

Persona que utiliza los servicios de un profesional o de una empresa, especialmente la que lo hace regularmente.

Framework

Es una estructura conceptual y tecnológica de asistencia definida con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software.

Calidad

Conjunto de propiedades inherentes a una cosa que permite caracterizarla y valorarla con respecto a las restantes de su especie.

Proceso

Conjunto de fases sucesivas de un fenómeno o hecho complejo.

Servicio

Organización, con su personal y medios, que se encarga de realizar un trabajo que satisface determinadas necesidades de una comunidad.

Arduino

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.

API

La interfaz de programación de aplicaciones es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

MATLAB

(Matrix Laboratory, “laboratorios de matrices”). Software matemático con entorno de desarrollo integrado (IDE) que tiene un lenguaje de programación propio (Lenguaje M) y es multiplataforma (Unix, Windows y Apple Mac Os X). Software de un gran uso en Centros de Investigación y Desarrollo, así como en universidades.

4.3. MARCO ESPACIAL

El proyecto se realizará desde casa permitiendo el trabajo remoto por la contingencia sanitaria que se vive en el país. Se contactará a las personas encargadas de suministrar la información y se harán reuniones por medios digitales de las personas involucradas en el proceso.

4.4. MARCO TEMPORAL

El desarrollo del proyecto de investigación se inicia en el mes de septiembre de 2020 y se tiene previsto su culminación en los próximos seis (6) meses.

Ilustración 1. Cronograma de actividades

Actividad	Septiembre				Octubre				Noviembre				Diciembre				Enero				Febrero			
Semanas	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Recopilar información, organizarla de forma detallada y delimitar el alcance y los requerimientos del sistema.																								
Investigar las opciones más viables para desarrollar la aplicación por medio de lenguajes de programación, frameworks y sistemas y elementos para la geolocalización de los buses.																								
Desarrollar un prototipo funcional de la aplicación móvil.																								
Desarrollar un prototipo de dispositivo de visualización que permita ver, al usuario, el tiempo de espera de su transporte de interés.																								
Efectuar pruebas en el prototipo de aplicativo móvil.																								

Fuente: El autor

5. METODOLOGÍA

5.1. TIPO DE INVESTIGACIÓN

Conjunto de buenas prácticas para trabajar colaborativamente en equipo y obtener el mejor resultado posible de proyectos, caracterizado por:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados.
- Solapar las diferentes fases del desarrollo.

Preferiblemente, es mejor utilizar la metodología ágil scrum, se ha visto y comprobado que es una manera efectiva para desarrollar este tipo de actividades; además, es ampliamente utilizada en empresas de software y es perfecta para un equipo de trabajo, donde se desarrollan diferentes roles y con el acompañamiento de varios tutores.

Scrum, la metodología Scrum es un marco de trabajo o framework que se utiliza dentro de equipos que manejan proyectos complejos. Es decir, se trata de una metodología de trabajo ágil que tiene como objetivo la entrega de valor en periodos cortos de tiempo y para ello se basa en tres pilares: la transparencia, inspección y adaptación. Esto permite al cliente, junto con su equipo comercial, insertar el producto en el mercado lo más rápido posible. Para este caso un aplicativo móvil.

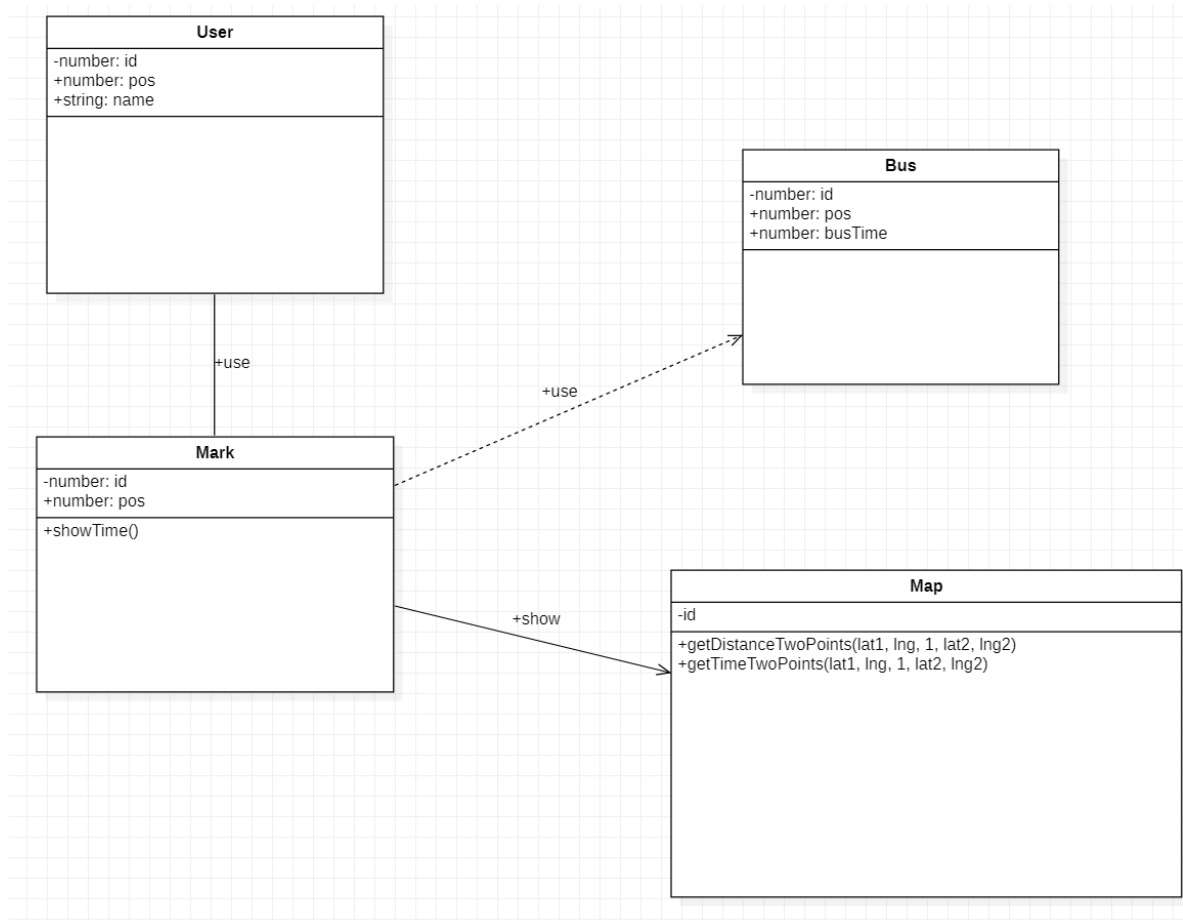
6. DESARROLLO DEL PROYECTO

Para el desarrollo del prototipo es importante tener como base una clase Map donde se encontrarán todas las modificaciones que se puedan realizar al mapa que provee la API de Google Maps.

Se cuenta con 4 clases:

- Usuario
- Marca (paradero)
- Bus
- Mapa

Ilustración 2. UML Prototipo Bus a Tiempo



Fuente: El autor

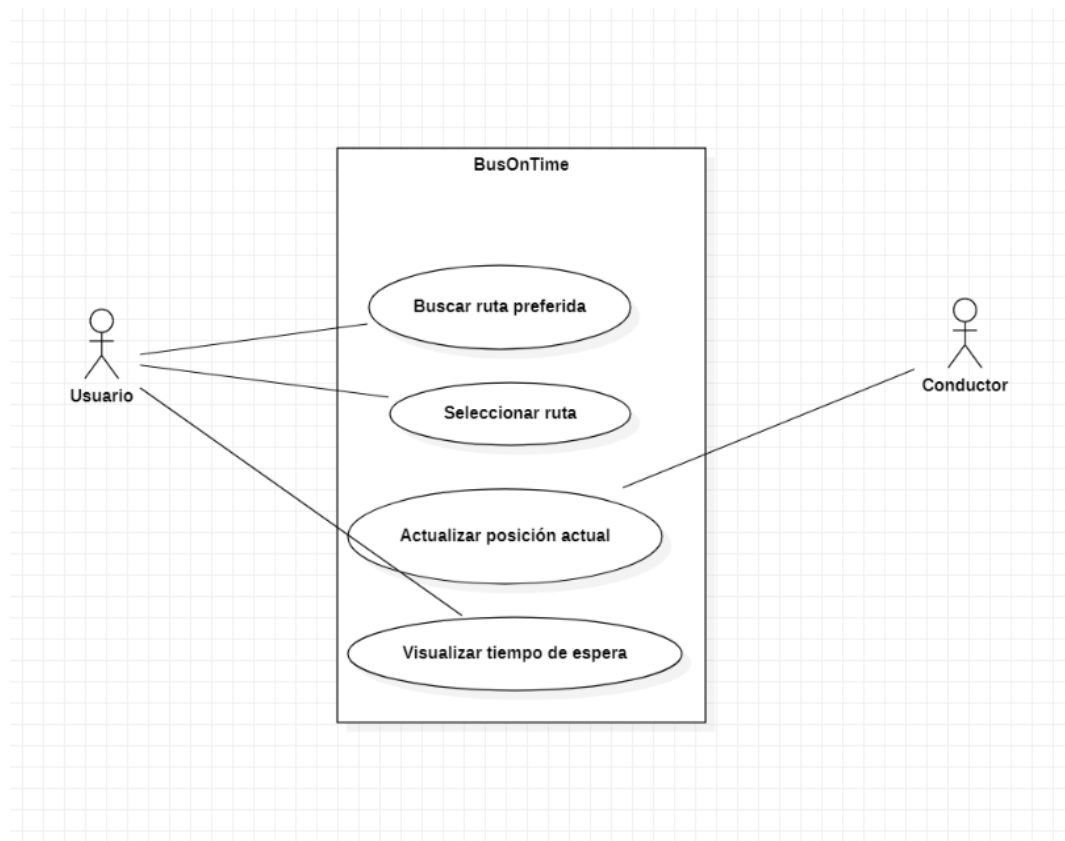
El usuario solo visualiza los valores que aparecen en la app, para este caso específico de una sola ruta y de distintos paraderos, al igual que puede visualizar en el mapa la posición del bus.

Hay paraderos donde solo se necesita la ubicación para poder luego poder calcular el tiempo de espera dependiendo de la posición del bus.

El bus comparte en tiempo real su posición actual, con la API de geolocation proporcionada por Google Maps se puede calcular su posición en pocos segundos, aunque, su efectividad depende del navegador, ya que el servicio se proporciona de esa forma.

El mapa que entraría a ser la clase principal, muestra todo lo referenciado anteriormente, desde algunas imágenes de presentación y también los tiempos de espera aproximados de la llegada del bus.

Ilustración 3. Caso de Uso Bus a Tiempo



Fuente: El autor

Es realmente sencillo lo que hace la aplicación y ya que es un prototipo el objetivo es mostrar la funcionalidad para luego ser escalada si debe ser necesario.

El usuario solo debe visualizar el tiempo de espera, puede ser de dos opciones, en la pantalla del aplicativo o en la pantalla LCD configurada por el Arduino UNO.

6.1. APLICACIÓN DESARROLLADA

Para el desarrollo del aplicativo se usa como Framework Angular 11, y diversas librerías de Angular Google Maps, ya que se usa la API que Google provee para poder usar su mapa. Angular provee una librería llamada AgmMap, y se usa para poder poner los componentes de HTML para la presentación de la página combinada con el código en TypeScript donde se encuentra la lógica del programa.

Para el ejemplo del prototipo se usa una parte de la ruta 27 del MegaBús (alimentador) que se encuentra en la página de Megabús donde se utiliza algunos puntos como paraderos donde se mide el tiempo de llegada del Bus a varios paraderos que se tienen puestos para el prototipo. Hablando de la fórmula para medir la distancia se usa la fórmula de Harvesine, que determina la distancia del círculo máximo entre dos puntos en una esfera dada sus longitudes y latitudes, para este caso el radio de la Tierra, y con la distancia calculada dependiendo de los puntos que se quieran medir, el cual no es exacta completamente, pero da un aproximado del tiempo de espera, qué es lo que se busca con esta implementación.

Se usa como base para medir el tiempo de espera una velocidad promedio de 30 Km/h, y con base a eso se mide el tiempo de espera con fracciones de la ruta.

Como base de la localización del mapa se utiliza la ubicación actual del bus con la API geolocation, la cual captura la latitud y la longitud captada por el hardware, es importante que el dispositivo tenga la posibilidad de compartir la ubicación, esta se irá actualizando a medida que el dispositivo se vaya desplazando.

Para tener un respaldo de la información recopilada, se usa Firebase que es proporcionada por Google.

Múltiples librerías fueron instaladas para poder ser utilizadas en el proyecto:

- NgModule
- BrowserAnimationsModule
- MatToolbarModule
- MatListModule
- MatGridListModule
- MatCardModule
- MatButtonModule

- GoogleMapModule
- AgmCoreModule (se dibuja el mapa)
- AgmDirectionModule
- AngularFireModule
- AngularFireDatabaseModule

Estas son las más importantes, aunque cabe recalcar que AgmDirectionModule el cual es el idóneo para dibujar el mapa en la ciudad con solo usar el AgmCoreModule y algunos puntos el mapa, pero se presentan problemas para ser cargado correctamente, así que, es preferible usar AgmPolyline aunque no sea de la mejor manera pero sigue siendo funcional.

Y por último se aprecia el tiempo de espera de los paraderos que están puestos en el mapa, cabe recalcar de nuevo que es un aproximado.

Para un desarrollo más completo de la aplicación se toma en cuenta varios aspectos que se usan en distintas herramientas que ya están en el mercado, como son: siniestros viales, demoras en el servicio, ruta preferida para un lugar determinado, información de los conductores y lugares famosos para visitar en la ciudad donde se desarrolle.

6.2. DISPOSITIVO DE VISUALIZACIÓN

6.2.1. Programa ARDUINO

La visualización del tiempo de espera en las estaciones o puntos de interés designados, se realiza a través de un dispositivo desarrollado con la plataforma Arduino.

Se escoge un software y hardware libre, del cual se encuentra mucha información y documentación disponible en la web, además de facilitar la interacción con otros programas que eran requeridos en el proyecto.

Se inicia por el montaje del circuito para el dispositivo que permitirá la visualización del tiempo de espera, con los siguientes elementos:

- Arduino Uno
- Display LCD 16x2
- Protoboard
- Potenciómetro
- Resistencia
- Cables

La conexión del circuito es sencilla y se basa en conocimiento adquirido durante la carrera, además de información disponible en la red. (*Ver anexo B*)

Lo siguiente es la implementación del código que permite controlar el display y la recepción y visualización de la variable (Tiempo de espera del usuario).

Esto se logra con la incorporación de la librería LiquidCrystal.h al código, que permite controlar el display a través de varias funciones específicas para este tipo de dispositivos. Además, se agregan funciones que permiten la lectura del puerto serial, por donde se conecta el dispositivo y por donde se realiza la lectura de la variable proveniente de otro programa (MATLAB). (*Ver anexo B*)

Las funciones más relevantes dentro del código son:

- lcd.write que permite la escritura de la variable y el texto que se desea visualizar en la pantalla LCD.
- La función Serial.available permite saber cuándo ha llegado nueva información al puerto serial.
- Serial.read, lee la información del puerto serial.

6.2.2. Programa MATLAB

Para la lectura del archivo que contiene la variable y envío de ésta al dispositivo de visualización, se utiliza el programa MATLAB.

Se utiliza este programa por tener un conocimiento previo en él y el cual tiene herramientas que permiten la conexión con la plataforma ARDUINO.

Lo primero que se programa, es la lectura del archivo que contiene la variable (tiempo de espera del usuario); esto se consigue con la función fopen, que permite abrir el archivo .txt, que, para este caso, es quien contiene la variable.

La función fread permite leer el contenido del archivo y se almacena en una variable que se utilizará más adelante en el código. Se debe cerrar nuevamente el archivo y para esto se utiliza la función fclose.

La comunicación con el dispositivo de visualización se realiza a través del puerto serial; para eso, se utilizan las siguientes funciones:

- La función instrfindall dentro de la función delete permite encontrar y borrar las comunicaciones que se tengan con el dispositivo a través del puerto serial.
- La función serial permite establecer la comunicación con el dispositivo conectado al puerto serial indicado y con los parámetros descritos allí.

- Nuevamente se utilizan las funciones fopen y fclose, para iniciar y cerrar la comunicación respectivamente.
- La función fprintf permite enviar la variable con el contenido del archivo .txt, a través de la comunicación que se estableció con el puerto serial; el formato establecido para este caso es un string, designado por la cadena '%s' del código.

Con todo implementado e incorporado queda listo el dispositivo que permite la visualización de tiempo de espera del transporte de interés del usuario. (Ver *anexo B y D*)

6.3. ENCUESTA

Se realiza una evaluación del conocimiento y acogida que puede llegar a tener este tipo de aplicaciones dentro de la ciudad. Para ello, se realiza una encuesta con una población muestral de alrededor de 160 personas de distintas edades, profesiones y estratos sociales.

No se busca diferenciar entre personas, grupos de edades o demás, simplemente pretende mostrar si se tiene conocimiento de este tipo de aplicaciones, la utilidad percibida, la usabilidad que se le daría y el valor agregado que podría aportar en su día a día.

Se va a usar la fórmula para el cálculo de la muestra de poblaciones finitas, es decir se conoce el total de la población y se desea saber cuál es el total de personas se debe estudiar, con la siguiente fórmula:

Ilustración 4. Fórmula para hallar muestra en población finita

$$n = \frac{N * Z_{\alpha}^2 * p * q}{d^2 * (N - 1) + Z_{\alpha}^2 * p * q}$$

Donde:

- N = Total de la población
- Z_{α} = 1.96 al cuadrado (si la seguridad es del 95%)
- p = Proporción esperada (en este caso 5% = 0.05)
- q = 1 - p
- d = Precisión

Ilustración 5. Resultado de la fórmula de muestra en población finita

Población	3000
Muestra	159

$$n = \frac{N * Z^2 * p * q}{d^2 * (N - 1) + Z^2 * p * q}$$

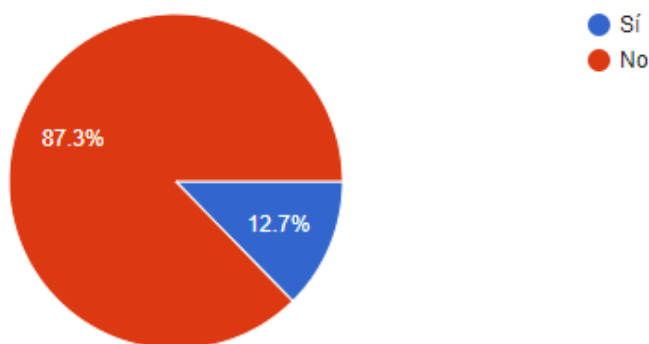
n	1037,232	132,24528
	7,843244	

Nivel de confianza	95% Z=1,96
--------------------	------------

d: precisión	5%
p	90%
q (1-p)	10%

Los resultados son los siguientes:

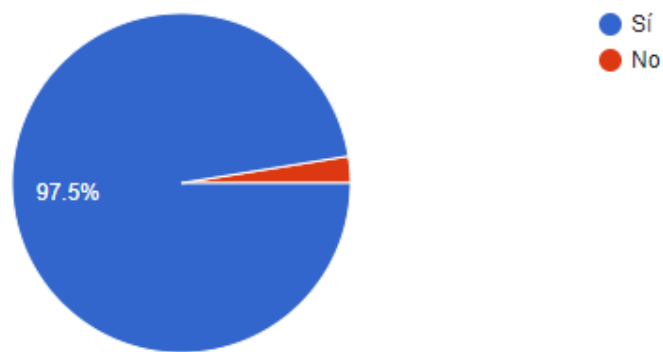
Ilustración 6. Respuestas de la pregunta número 1



Fuente: El autor

Esto demuestra que el 87.3% de las personas que respondieron, no conocen una aplicación para consultar la ruta y horarios de los buses de transporte público en su ciudad. Muestra que, en la mayoría de las ciudades del país aún no se tiene este tipo de sistemas implementados o completamente funcionales; también, puede demostrar una falta de información y motivación a los usuarios para el uso de estas aplicaciones.

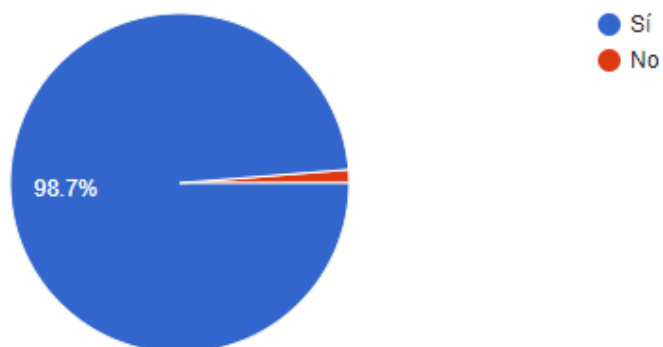
Ilustración 7. Respuestas de la pregunta número 2



Fuente: El autor

La estadística anterior, comprueba que la aplicación se percibe como una herramienta útil, pues el 97,5% de las personas encuestadas respondieron afirmativamente.

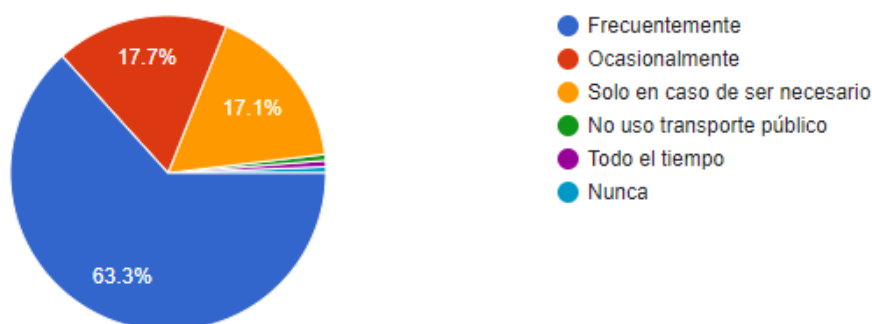
Ilustración 8. Respuestas de la pregunta número 3



Fuente: El autor

Que el 98,7% de las respuestas sean afirmativas, deja ver que el aplicativo, no solo se considera útil, sino también, una herramienta que genera un valor agregado, pues permitiría una mejor gestión del tiempo de los usuarios de transporte público.

Ilustración 9. Respuestas de la pregunta número 4



Fuente: El autor

Cuando se pregunta por la frecuencia de uso que se le daría al sistema, la mayoría de las personas le daría un uso recurrente; ya que, el 63,3% de las respuestas fueron por un uso frecuente de la aplicación.

Esta investigación demuestra que la creación de un sistema de información para la consulta de rutas y horarios de los buses de transporte público tiene una gran acogida; no solo se ve como una idea innovadora, sino útil, práctica y usable por parte de los usuarios. *(Ver anexo C)*

6.4. PRUEBAS

Se realizaron una serie de pruebas al prototipo de la aplicación desarrollada el día 26 de enero y el día 5 de febrero del 2021, se evalúa si el tiempo mostrado en pantalla, es equivalente al tomado por un cronómetro dentro de un vehículo a velocidad de 30 km/h, que es la velocidad programada en el código fuente. Las aproximaciones fueron positivas, aunque se tenían algunos inconvenientes con la ubicación proporcionada por la función de Api de Google maps Geolocation.

Las pruebas realizadas demuestran que la fórmula funciona bien, pero debe ser optimizada en la aplicación y en su llamado, y además se debe tener otra función más acertada para la ubicación.

En las pruebas realizadas el día 26 de enero se tuvo una aproximación cercana al tiempo mostrado por el prototipo, solo falla por 30 segundos, y en las pruebas realizadas el día 5 de febrero hay más precisión, ya que se ajustó la fórmula harvesine, pero igualmente se tuvieron algunos problemas con la API Geolocation.

(Ver anexo D y E)

7. CONCLUSIONES

- Se realiza una evaluación del estado actual de ideas similares en la ciudad de Pereira y algunos aplicativos funcionales en otras ciudades del país; además, se revisan antecedentes de la idea a nivel mundial. En los lugares que se encuentran implementados este tipo de sistemas, se destaca la importancia que tienen en la gestión y aprovechamiento del tiempo por parte de sus usuarios, además del mejoramiento del servicio de transporte público y la satisfacción de dichos usuarios.
- Se observa que la implementación de un sistema de información para la gestión de los datos relacionados con la consulta de rutas y horarios de los buses en la ciudad de Pereira, puede utilizar para su diseño de implementación herramientas como Node Js, Angular, Arduino o Matlab y API's como Google Maps API, API Directions o API Roads; estas herramientas son la base tanto para el desarrollo del sistema de información como para la implementación de los prototipos de la aplicación y dispositivo de visualización.
- Se evidencia que existen diferentes variables a las que se le debe dar un buen manejo y tratamiento en aspectos de visualización, entrega de información, geolocalización de los buses, puntos de interés, velocidad de los buses y demás datos; ya que deben estructurarse de acuerdo a la forma en que se va a procesar y visualizar la información.
- Apuntar a ser un modelo de ciudad más inteligente puede ser posible con la implementación correcta de este tipo de sistemas. El transporte público de la ciudad de Pereira, puede llegar a ser más organizado teniendo herramientas que le permitan al usuario de transporte público, saber el tiempo de llegada de su transporte de interés.

8. RECOMENDACIONES

- Es fundamental tener claridad en los objetivos y el trabajo que se pretende hacer, para llevar un control y seguimiento sistemático a los avances realizados.
- Es importante el conocimiento específico en el campo de la programación y especificidad de cada programa con el que se pretende trabajar; conocer funciones dedicadas y especializadas para realizar una tarea en concreto.
- La lógica de la programación requiere una buena capacidad de análisis y una secuencia de instrucciones muy ordenada, sistematizada y específica para que no haya cabida a errores. Además, se recomienda comentar los códigos, en el futuro se puede no recordar su funcionalidad o puede permitir una mejor interpretación de otras personas.
- El desarrollo del aplicativo se realiza sobre una ruta en concreto, para las demás rutas se recomienda replicar el formato, programación y demás elementos, funciones y aplicaciones que se dieron para la creación de ésta.
- El aplicativo para la consulta de rutas y horarios de los buses en la ciudad de Pereira y el dispositivo de visualización, son prototipos funcionales y tienen margen de mejora en aspectos como la programación, presentación, eficiencia, elementos y programas que se usaron para la creación de éstos.

BIBLIOGRAFÍA

Documento Bus a tiempo, desarrollado por el estudiante Juan Miguel Vargas Cortes.

Scrum Manager. Guía de formación Versión 2.6 [En línea], [Revisado 17 de Noviembre de 2020]. Disponible en internet:

https://www.scrummanager.net/files/sm_proyecto.pdf

Aplicación MyTransit: <https://mytransit.es.aptoide.com/app>

Rutas del servicio de transporte MEGABUS. [En línea], [Revisado 17 de noviembre de 2020]. Disponible en internet:

<https://www.megabus.gov.co/index.php/servicios/rutas/rutas-datos-abiertos>

Misión, visión y valores de la empresa de transporte MEGABUS. [En línea], [Revisado 17 de noviembre de 2020]. Disponible en internet:

<https://megabus.gov.co/index.php/ct-megabus/plataforma-estrategica>

Sistema de información. [En línea], [Revisado 16 de Noviembre de 2020]. Disponible en internet: <https://www.britannica.com/topic/information-system>

Piccolo, Gabriele; Pigni, Federico (July 2018). Information systems for managers: with cases (Edition 4.0 ed.). Prospect Press. p. 28.

O'Hara, Margaret; Watson, Richard; Cavan, Bruce (1999). "Managing the three levels of change". Information Systems Management. 16 (3): 64. [En línea], [Revisado 16 de noviembre de 2020]. Disponible en internet:

https://www.researchgate.net/publication/220630523_Managing_the_three_Levels_of_Change

¿Qué es Arduino? [En línea], [Revisado 18 de enero de 2021]. Disponible en internet: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

Historia de los sistemas de información [En línea], [Revisado 18 de enero de 2021]. Disponible en internet:

<https://blogs.deusto.es/master-informatica/un-recorrido-por-la-historia-de-los-si/>

MATLAB [En línea], [Revisado 25 de enero de 2021]. Disponible en internet:

<https://www.ecured.cu/MATLAB>

AngularJS [En línea], [Revisado 28 de enero de 2021]. Disponible en internet: <https://es.wikipedia.org/wiki/AngularJS>

Qué es frontend y backend [En línea], [Revisado 25 de enero de 2021]. Disponible en internet: <https://platzi.com/blog/que-es-frontend-y-backend/>

Modelo vista controlador [En línea], [Revisado 25 de enero de 2021]. Disponible en internet: [https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html#:~:text=Modelo%20Vista%20Controlador%20\(MVC\)%20es,control%20e n%20tres%20componentes%20distintos.&text=La%20Vista%2C%20o%20interfaz %20de,los%20mecanismos%20interacci%C3%B3n%20con%20%C3%A9ste.](https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html#:~:text=Modelo%20Vista%20Controlador%20(MVC)%20es,control%20e n%20tres%20componentes%20distintos.&text=La%20Vista%2C%20o%20interfaz %20de,los%20mecanismos%20interacci%C3%B3n%20con%20%C3%A9ste.)

¿Qué es firebase? [En línea], [Revisado 29 de enero de 2021]. Disponible en internet: <https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/>

¿Qué son google apis? [En línea], [Revisado 27 de enero de 2021]. Disponible en internet: <https://maplink.global/es/blog/que-son-google-apis/>

Metodología scrum [En línea], [Revisado 28 de enero de 2021]. Disponible en internet: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html#:~:text=Scrum%20es%20una%20metodolog%C3%ADa%20de,equi pos%20que%20manejan%20proyectos%20complejos>

ANEXOS

ANEXO A. Aplicación Desarrollada

Ilustración 1. Interfaz Bus a Tiempo



Ilustración 2. Mapa Api Google Maps

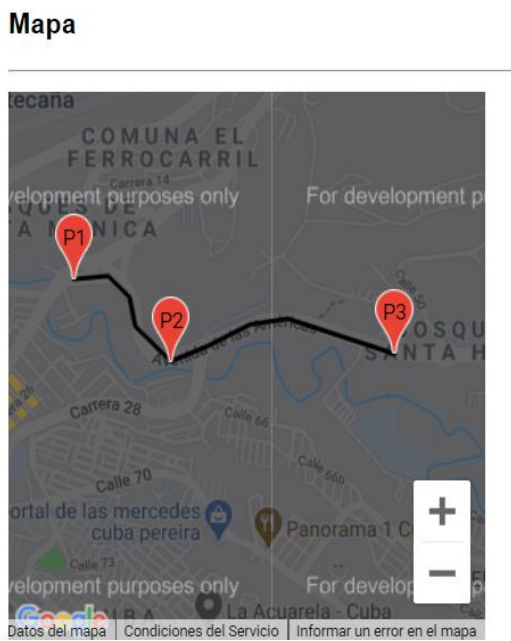


Ilustración 3. Tabla de tiempos

TIEMPO
Paradero 1 Tiempo: 6m:39s
Paradero 2 Tiempo: 6m:6s
Paradero 3 Tiempo: 8m:31s
Export To json
Export To txt

Ilustración 4. Función para obtener la distancia

```
getDistance(pos1,pos2){
  let R = 6371;
  let lat = this.radio(pos2[0] - pos1[0]);
  let lng = this.radio(pos2[1] - pos1[1]);
  let a = angularMath.sinNumber(lat/2) * angularMath.sinNumber(lat/2) + angularMath.cosNumber(this.radio(pos1[0])) * angularMath.cosNumber(this.radio(pos2[0])) * angularMath.sinNumber(lng / 2) * angularMath.sinNumber(lng / 2);
  let c1 = angularMath.atanNumber(angularMath.squareOfNumber(a));
  let c2 = angularMath.atanNumber(angularMath.squareOfNumber(1-a));
  let c3 = angularMath.diu(c1,c2);
  let c = 2 * angularMath.atanNumber(c3);
  let d = R * c;
  //console.log("Distance = " + d);
  return d;
}
```

Ilustración 5. Convertir el número resultante de tiempo a horas, minutos y segundos

```
getTimeMinute(num):Array<number>{
  let hora = parseInt((num).toFixed(0));
  let minuto = ((num%1)*60);
  let segundo = parseInt(((minuto%1)*60).toFixed(0));
  // console.log("h: " + hora + " m: " + minuto.toFixed(0) + " s: " + segundo);
  let v : Array<number>
  v= new Array<number>()
  v.push(hora);
  v.push(minuto);
  v.push(segundo);
  return v;
}
```

Ilustración 6. Función para obtener el tiempo

```
getTime(dis1,dis2) {  
  let distance = this.getDistance([dis1[0],dis1[1]],[dis2[0],dis2[1]]);  
  let time = distance/this.speed;  
  console.log(time);  
  let v7 = this.getTimeMinute(time);  
  console.log(v7[0]);  
  
  return (" Tiempo: " + v7[1].toFixed(0) + "m:" + v7[2] + "s");  
}
```

ANEXO B. Dispositivo de visualización

Ilustración 1. Esquema de conexión del dispositivo de visualización

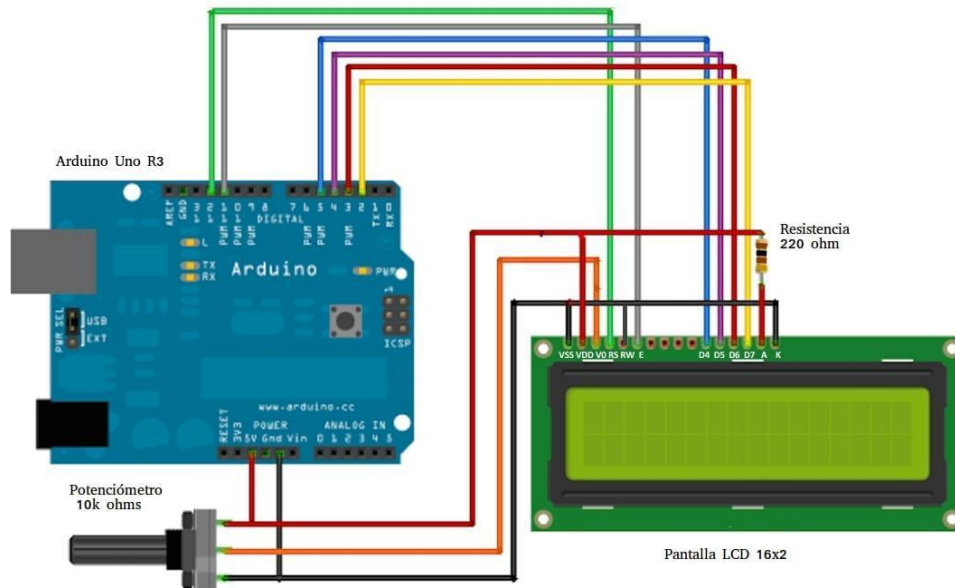


Ilustración 2. Dispositivo de visualización.

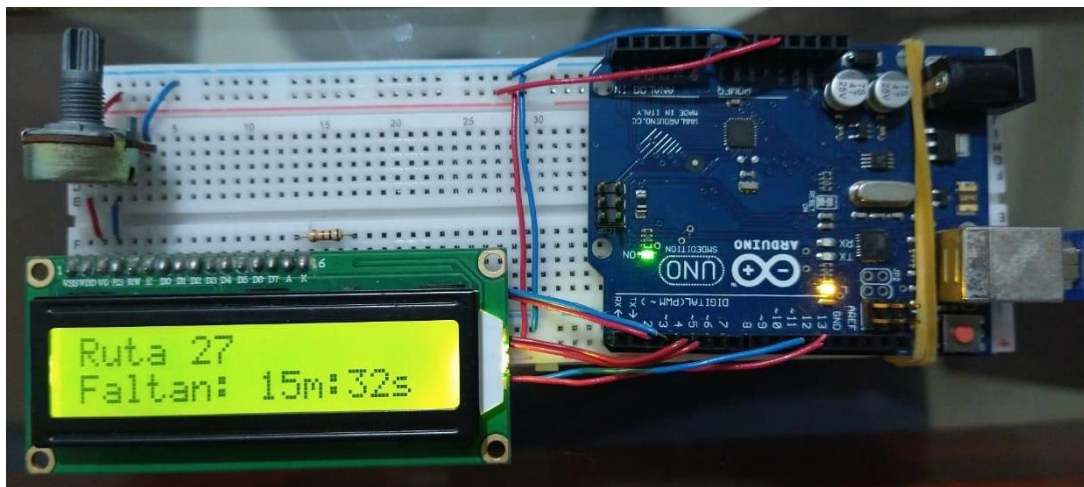


Ilustración 3 y 4. Código de Arduino

VisualizarTiempo Arduino 1.8.12

Archivo Editar Programa Herramientas Ayuda

VisualizarTiempo

```

1 //incluye libreria para controlar la LCD
2 #include <LiquidCrystal.h>
3
4 // Configuración de los pines para la comunicación con la LCD
5 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
6
7 void setup()
8 {
9     //Se inicia el puerto serial
10    Serial.begin(9600);
11
12    //Indicamos que tenemos conectada una pantalla de 16x2
13    lcd.begin(16, 2);
14
15    //Configuración inicial del display
16    lcd.home();
17    lcd.write("Ruta 27");
18    lcd.setCursor(0,1);
19    lcd.write("Faltan: ");
20
21 }
22
23 void loop()
24 {
25
26     //Cuando los caracteres lleguen al puerto serial...
27     if (Serial.available()) {
28
29         //Espera para que llegue el mensaje entero
30         delay(100);
31
32         lcd.clear();
33         lcd.home();
34         lcd.write("Ruta 27");
35         lcd.setCursor(0,1);
36         lcd.write("Faltan: ");
37         lcd.setCursor(8,1);
38
39         //lee todos los caracteres disponibles
40         while (Serial.available() > 0) {
41
42             //Muestra cada caracter en la LCD
43             lcd.write(Serial.read());
44         }
45     }
46
47 }

```

VisualizarTiempo Arduino 1.8.12

Archivo Editar Programa Herramientas Ayuda

VisualizarTiempo

```

19    lcd.write("Faltan: ");
20
21 }
22
23 void loop()
24 {
25
26     //Cuando los caracteres lleguen al puerto serial...
27     if (Serial.available()) {
28
29         //Espera para que llegue el mensaje entero
30         delay(100);
31
32         lcd.clear();
33         lcd.home();
34         lcd.write("Ruta 27");
35         lcd.setCursor(0,1);
36         lcd.write("Faltan: ");
37         lcd.setCursor(8,1);
38
39         //lee todos los caracteres disponibles
40         while (Serial.available() > 0) {
41
42             //Muestra cada caracter en la LCD
43             lcd.write(Serial.read());
44         }
45     }
46
47 }

```

Ilustración 5. Código de Matlab

LecturaEnvioTxt.m x hola.txt x +

```

1 - clear all;
2 - clc;
3
4 - fileID=fopen('hola.txt','r'); % abre el archivo .txt
5 - A = fread(fileID,'*char'); % Lee el contenido del archivo
6 - fclose(fileID); % cierra el archivo
7
8 - delete(instrfindall({'Port'},{'COM3'})); % borra comunicaciones que existan en el puerto
9 - arduino=serial('COM3','BaudRate',9600); % crea una comunicacion serial con el puerto
10 - fopen(arduino); % inicia comunicacion
11 - pause(2);
12 - fprintf(arduino,'%s',A); % Lo que se quiere transmitir
13 - fclose(arduino); % cierra comunicacion
14

```


ANEXO C. Encuesta

Ilustración 1. Pregunta número 1

¿Conoce alguna aplicación para consultar las rutas y horarios del transporte público en su ciudad?

☐ Sí

☐ No

Ilustración 2. Pregunta número 2

¿Consideraría útil, una aplicación que le permita ver, cuanto tiempo falta para que su transporte pase por el lugar en que usted se encuentra?

☐ Sí

☐ No

Ilustración 3. Pregunta número 3

¿Cree usted que, una aplicación que permita consultar las rutas, horarios y le indique el tiempo de llegada de su transporte, le permitiría gestionar mejor su tiempo a lo largo del día?

☐ Sí

☐ No

Ilustración 4. Pregunta número 4

Como usuario recurrente del transporte público ¿Con que frecuencia usaría este tipo de aplicativos?

- ☐ Frecuentemente
- ☐ Ocasionalmente
- ☐ Solo en caso de ser necesario
- ☐ Otra...

ANEXO D. Videos

Enlace video introductorio:

<https://www.youtube.com/watch?v=QqtglLOCHnE>

Enlace video de prueba del aplicativo:

<https://www.youtube.com/watch?v=vrfp3k6tKso>

Enlace video de prueba del dispositivo de visualización:

<https://www.youtube.com/watch?v=TmsWZkCxe6s>

ANEXO E. Tabla de Pruebas

Toma de pruebas 26 de enero

Posición	Tiempo Real	Tiempo Estimado
Evaluación 1	3m:27s	2m:55s

Toma de pruebas 5 de febrero

Posición	Real	Tiempo Estimado
Evaluación 1	1m:56s	2m:2s
Evaluación 2	1m:40s	1m:25s
Evaluación 3	0m:13s	0m:11s

ANEXO F. Respuesta a la solicitud de información de la empresa MEGABÚS S.A.

A continuación, se detalla el contenido de la respuesta a la solicitud de información por parte de la empresa MEGABÚS S.A.:

Señores

Juan Pablo Botero Bermúdez

Juan Miguel Vargas Cortés

Estudiantes UTP

boterobpablo@utp.edu.co

Asunto: Respuesta PQRS No. 608 “Solicitud información para trabajo de grado”.

1- Certificación de si tiene o no una aplicación completamente funcional donde los usuarios del servicio pueden consultar las rutas, horarios y tiempos de espera de los buses o articulados en Pereira, en el momento de hacer esta solicitud.

MEGABUS S.A, no cuenta con una aplicación tecnológica diseñada para consulta en tiempo real de tiempos de espera de nuestros vehículos, horarios y rutas del SITM.

Sin embargo, los usuarios del sistema MEGABUS pueden consultar las rutas (alimentadoras y troncales) y recorridos, en la página de la entidad www.megabus.gov.co.

2- Número promedio de pasajeros que se movilizan diariamente utilizando el servicio, con números antes de la pandemia y cuarentenas.

La demanda promedio de pasajeros en el Sistema Integrado de Transporte Masivo MEGABUS es de 120.000 día hábil, antes de la pandemia por Covid-19.

3- Toda la información referente a la ruta No. 27 del servicio de alimentadores, número promedio de pasajeros que se movilizan diariamente, longitud de la ruta, frecuencia de despacho y horarios, igualmente con cifras antes de la pandemia.

***No. Promedio de pasajeros:** La demanda promedio de pasajeros de la ruta No 27 - UTP, es de 3.000 día hábil, antes de la pandemia por Covid-19.

***Longitud de la ruta UTP:** 17.30 km

***Frecuencias y horarios Ruta 27:** Se adjuntan al presente oficio (Día hábil, sábados, Domingos y festivos) en condiciones normales, sin pandemia.

Cabe aclarar que el Plan de Servicios de cada ruta está sujeto a modificación y ajustes, dependiendo de la variación en la demanda.

Cordial saludo.

Contáctenos MEGABUS.

Igualmente se agrega el enlace para consultar el documento oficial:

https://drive.google.com/file/d/10BxukiALP3xNiOdrULU3gl23d3J55bkf/view?usp=s_haring

A continuación, se describe la información detallada de la ruta número 27 del alimentador de MEGABUS:

RUTA 27 HÁBILES			RUTA 27 HÁBILES		
BARRIO	INTER. CUBA	tabla	BARRIO	INTER. CUBA	tabla
4:27	4:55	1	8:18	8:46	3
	5:00	2	8:24	8:51	4
	5:06	3	8:29	8:57	5
	5:11	4	8:35	9:02	6
4:49	5:17	5	8:40	9:08	7
4:55	5:22	6	8:46	9:13	8
5:00	5:28	7	8:51	9:19	9
5:06	5:33	8	8:57	9:24	10
5:11	5:39	9	9:02	9:30	1
5:17	5:44	10	9:08		2
5:22	5:50	1	9:13	9:41	3
5:28	5:55	2	9:19		4
5:33	6:01	3	9:24	9:52	5
5:39	6:06	4	9:30		6
5:44	6:12	5	9:35	10:03	7
5:50	6:17	6	9:41		8
5:55	6:23	7	9:46	10:14	9
6:01	6:28	8	9:52		10
6:06	6:34	9	9:57	10:25	1
6:12	6:39	10	10:08	10:36	3
6:17	6:45	1	10:19	10:47	5
6:23	6:50	2	10:30	10:58	7
6:28	6:56	3	10:41	11:09	9
6:34	7:01	4	10:52	11:20	1
6:39	7:07	5	11:03	11:31	3
6:45	7:12	6	11:14	11:42	5
6:50	7:18	7	11:25	11:53	7
6:56	7:23	8	11:36	12:04	9
7:01	7:29	9	11:47	12:15	1
7:07	7:34	10	11:58	12:26	3
7:12	7:40	1	12:09	12:37	5
7:18	7:45	2	12:20	12:48	7
7:23	7:51	3	12:31	12:59	9
7:29	7:56	4	12:42	13:10	1
7:34	8:02	5	12:53	13:21	3
7:40	8:07	6	13:04	13:32	5
7:45	8:13	7	13:15	13:43	7
7:51	8:18	8	13:26	13:54	9
7:56	8:24	9	13:37	14:05	1
8:02	8:29	10	13:48	14:16	3
8:07	8:35	1	13:59	14:27	5
8:13	8:40	2	14:10	14:38	7

RUTA 27 HÁBILES			RUTA 27 HÁBILES		
BARRIO	INTER. CUBA	tabla	BARRIO	INTER. CUBA	tabla
14:21	14:49	9	18:40	19:07	8
14:32	15:00	1	18:45	19:13	7
14:43	15:11	3	18:51	19:18	10
14:54	15:22	5	18:56	19:24	9
15:05	15:33	7	19:02	19:29	2
15:16	15:44	9	19:07	19:35	1
15:22	15:49	2	19:13	19:40	4
15:27	15:55	1	19:18	19:46	3
15:33	16:00	4	19:24	19:51	6
15:38	16:06	3	19:29	19:57	5
15:44	16:11	6	19:35	20:02	8
15:49	16:17	5	19:40	20:08	7
15:55	16:22	8	19:46	20:13	10
16:00	16:28	7	19:51	20:19	9
16:06	16:33	10	19:57	20:24	2
16:11	16:39	9	20:02	20:30	1
16:17	16:44	2	20:08	20:35	4
16:22	16:50	1	20:13	20:41	3
16:28	16:55	4	20:19	20:46	6
16:33	17:01	3	20:24	20:52	5
16:39	17:06	6	20:30	20:57	8
16:44	17:12	5	20:35	21:03	7
16:50	17:17	8	20:41	21:08	10
16:55	17:23	7	20:46	21:14	9
17:01	17:28	10	20:52	21:19	2
17:06	17:34	9	20:57	21:25	1
17:12	17:39	2	21:03	21:30	4
17:17	17:45	1	21:08	21:36	3
17:23	17:50	4	21:14	21:41	6
17:28	17:56	3	21:19	21:47	5
17:34	18:01	6	21:25	21:52	8
17:39	18:07	5	21:30	21:58	7
17:45	18:12	8	21:36	22:03	10
17:50	18:18	7	21:41	22:09	9
17:56	18:23	10	21:47	22:14	2
18:01	18:29	9	21:52	22:20	1
18:07	18:34	2	22:03	22:31	3
18:12	18:40	1	22:14	22:42	5
18:18	18:45	4	22:25	22:53	7
18:23	18:51	3	22:36	23:04	9
18:29	18:56	6	22:47		1
18:34	19:02	5	22:58		3
			23:09		5

RUTA 27 SÁBADOS		
BARRIO	INTER. CUBA	tabla
	4:55	1
	5:04	2
	5:13	3
4:55	5:22	4
5:04	5:31	5
5:13	5:40	6
5:22	5:49	1
5:31	5:58	2
5:40	6:07	3
5:49	6:16	4
5:58	6:25	5
6:07	6:34	6
6:16	6:43	1
6:25	6:52	2
6:34	7:01	3
6:43	7:10	4
6:52	7:19	5
7:01	7:28	6
7:10	7:37	1
7:19	7:46	2
7:28	7:55	3
7:37	8:04	4
7:46	8:13	5
7:55	8:22	6
8:04	8:31	1
8:13	8:40	2
8:22	8:49	3
8:33	9:00	4
8:44	9:11	5
8:55	9:22	6
9:06	9:33	2
9:17	9:44	3
9:28	9:55	4
9:39	10:06	5
9:50	10:17	6
10:01	10:28	2
10:12	10:39	3
10:23	10:50	4
10:34	11:01	5
10:45	11:12	6
10:56	11:23	2
11:07	11:34	3
11:18	11:45	4
11:29	11:56	5
11:40	12:07	6
11:51	12:18	2
12:02	12:29	3
12:13	12:40	4
12:24	12:51	5
12:35	13:02	6
12:46	13:13	2
12:57	13:24	3
13:08	13:35	4
13:19	13:46	5

RUTA 27 SÁBADOS		
BARRIO	INTER. CUBA	tabla
13:30	13:57	6
13:41	14:08	2
13:52	14:19	3
14:03	14:30	4
14:14	14:41	5
14:25	14:52	6
14:36	15:03	2
14:47	15:14	3
14:58	15:25	4
15:09	15:36	5
15:20	15:47	6
15:31	15:58	2
15:42	16:09	3
15:53	16:20	4
16:04	16:31	5
16:15	16:42	6
16:26	16:53	2
16:37	17:04	3
16:48	17:15	4
16:59	17:26	5
17:10	17:37	6
17:21	17:48	2
17:32	17:59	3
17:43	18:10	4
17:54	18:21	5
18:05	18:32	6
18:16	18:43	2
18:27	18:54	3
18:38	19:05	4
18:49	19:16	5
19:00	19:27	6
19:11	19:38	2
19:22	19:49	3
19:33	20:00	4
19:44	20:11	5
19:55	20:22	6
20:06	20:33	2
20:17	20:44	3
20:28	20:55	4
20:39	21:06	5
20:50	21:17	6
21:01	21:28	2
21:12	21:39	3
21:23	21:50	4
21:34	22:01	5
21:45	22:12	6
21:56	22:23	2
22:07	22:34	3
22:18	22:45	4
22:31	22:56	5
22:39		6
22:50		2
23:01		3

RUTA 27 DOMINGO			RUTA 27 DOMINGO		
INTER. CUBA	tabla	BARRIO	INTER. CUBA	tabla	BARRIO
	5:30	1	13:57	14:23	2
	5:43	2	14:10	14:36	3
	5:56	3	14:23	14:49	4
5:43	6:09	4	14:36	15:02	1
5:56	6:22	1	14:49	15:15	2
6:09	6:35	2	15:02	15:28	3
6:22	6:48	3	15:15	15:41	4
6:35	7:01	4	15:28	15:54	1
6:48	7:14	1	15:41	16:07	2
7:01	7:27	2	15:54	16:20	3
7:14	7:40	3	16:07	16:33	4
7:27	7:53	4	16:20	16:46	1
7:40	8:06	1	16:33	16:59	2
7:53	8:19	2	16:46	17:12	3
8:06	8:32	3	16:59	17:25	4
8:19	8:45	4	17:12	17:38	1
8:32	8:58	1	17:25	17:51	2
8:45	9:11	2	17:38	18:04	3
8:58	9:24	3	17:51	18:17	4
9:11	9:37	4	18:04	18:30	1
9:24	9:50	1	18:17	18:43	2
9:37	10:03	2	18:30	18:56	3
9:50	10:16	3	18:43	19:09	4
10:03	10:29	4	18:56	19:22	1
10:16	10:42	1	19:09	19:35	2
10:29	10:55	2	19:22	19:48	3
10:42	11:08	3	19:35	20:01	4
10:55	11:21	4	19:48	20:14	1
11:08	11:34	1	20:01	20:27	2
11:21	11:47	2	20:14	20:40	3
11:34	12:00	3	20:27	20:53	4
11:47	12:13	4	20:40	21:06	1
12:00	12:26	1	20:53	21:19	2
12:13	12:39	2	21:06	21:32	3
12:26	12:52	3	21:19	21:45	4
12:39	13:05	4	21:32	21:58	1
12:52	13:18	1	21:45	22:11	2
13:05	13:31	2	21:58	22:24	3
13:18	13:44	3	22:11	22:35	4
13:31	13:57	4	22:24		1
13:44	14:10	1	22:35		2